

Image Recognition of Disease-Carrying Insects: A System for Combating Infectious Diseases Using Image Classification Techniques and Citizen Science

J. Pablo Muñoz
City University of New York
New York, NY, USA
jmunoz2@gradcenter.cuny.edu

Rebecca Boger, Scott Dexter
and Justin Li
Brooklyn College
Brooklyn, NY, USA
{rboger, sdexter, lj0831}
@brooklyn.cuny.edu

Russanne Low
Institute for Global
Environmental Strategies
Arlington, VA, USA
rusty_low@strategies.org

Abstract

We propose a system that assists infectious disease experts in the rapid identification of potential outbreaks resulting from arboviruses (mosquito, ticks, and other arthropod-borne viruses). The proposed system currently identifies mosquito larvae in images received from citizen scientists. Mosquito-borne viruses, such as the recent outbreak of Zika virus, can have devastating consequences in affected communities. We describe the first implemented prototype of our system, which includes modules for image collection, training of image classifiers, specimen recognition, and expert validation and analytics. The results of the recognition of specimens in images provided by citizen scientists can be used to generate visualizations of geographical regions of interest where the threat of an arbovirus may be imminent. Our system uses state-of-the-art image classification algorithms and a combination of mobile and desktop applications to ensure that crucial information is shared appropriately and accordingly among its users.

1. Introduction

Recently, several regions of the world, including the Americas, Africa and parts of Asia, were on the brink of a Zika virus epidemic. The crisis has not completely abated, and there are still chances of the virus reappearing in vulnerable communities. Zika virus, or any other arbovirus, can have devastating consequences. Many of the communities most affected by Zika and other arboviruses do not have the capacity to efficiently monitor and track these diseases. Public health officials find themselves stymied when confronted by potential outbreaks because the speed required to react to those potential epidemics can be very difficult where a lack of infrastructure and effective means of communicating with the public exist. In the case of Zika, only after thousands of cases of microcephaly were reported was a coordinated re-

sponse from government officials put in place. Most nations, even among developed countries, have found preparing to confront an arbovirus epidemic difficult [1].

Communities at risk for an arbovirus outbreak can benefit greatly from technology that allows them to interact directly with mosquito experts. Public health officials can also benefit from alert mechanisms and better data analysis and prediction, which could provide real-time dynamic updates on the current and predicted trajectory of the epidemic. This kind of collaboration between the general public and scientists is the basis of “citizen science.” In this paper, we present a system that uses *Deep Learning* image classification techniques and citizen science to assist communities in detecting dangerous mosquito larvae. While we have focused on Zika virus, the technology described in this document could also be customized to respond to other mosquito-borne epidemics, such as encephalitis, dengue and yellow fever, accelerating the public health response. The system could also be tailored to confront other problems in which image classification and mobile applications can be used by citizen scientists.

Citizen science and crowdsourcing are important means of deterring the spread of disease. With state-of-the-art technology, these approaches can greatly increase and improve their positive contributions. One of our goals is to provide citizen scientists with access to the latest technologies for mobile communication, image classification and object recognition, in addition to information about their potential uses, so they can better help their communities. The optical recognition system developed here is an important element of streamlining the data quality/data assurance process. This is particularly important for a citizen science project that is global in scope with the potential to have thousands of people submitting photo data.

Our prototype is an innovative solution that follows the premises of the *Design Science* paradigm [2], in which novel artifacts help improve *Information Systems (IS)*. Our pro-

prototype can be further improved and we are convinced that researchers and public health experts will find its potential utility in real-world applications. The ongoing improvement of the system can also benefit from *Action Design Research* (ADR) [3] in which iterations of concurrent development and user evaluation of the system will help us identify emergent behavior that may not have been anticipated during the design of our prototype.

This paper is organized as follows: Section 2 presents related work in image classification and the use of technology in citizen science. Section 3 gives an overview of the proposed system. Section 4 describes the first prototype implemented as a proof of concept of the system. Section 5 shows initial results of our experiments with the first prototype of the system. Section 6 presents future work to be done on the proposed system.

2. Related Work

Image classification refers to the problem of assigning a label to a provided image. Image classification approaches have reached a very high accuracy in their predictions in the past few years. Until 2011, the error rate in the *ImageNet Large-Scale Visual Recognition Challenge*—one of the most popular international image recognition competitions—was about 25%. In 2012, a participant using an *Artificial Neural Network* approach known as *Deep Learning* [4] won this challenge for the first time, obtaining an outstanding error rate of 16% [5], significantly outperforming traditional methods for object recognition. Since that time, more teams using *Deep Learning* approaches have entered the competition, reducing the error rate to the single digits [6].

Broadly speaking, *Deep Learning* refers to the use of *Artificial Neural Networks* (ANN) with a great number of layers. It has been used for a few decades in everyday applications. For instance, in its earliest uses, *Deep Learning* was applied to document recognition [7]. The revival and relatively recent success of this approach is due mostly to Moore's Law [8] and to the availability of more powerful computers with Graphics Processing Units (GPU) that allow them to perform many more operations in parallel than Central Processing Units (CPU). In addition to the improvements in GPUs, there have been several refinements to traditional *Artificial Neural Networks* implementations that have resulted in better classification schemes [9] [10].

Over the past 20 years, the number of citizen science projects and the number of people participating in these projects have risen dramatically [11]. In 2014, the phrase "citizen science" was becoming so commonplace in usage that it was included in the Oxford English dictionary, which defines citizen science as "[t]he collection and analysis of data relating to the natural world by members of the general public, typically as part of a collaborative project with professional scientists" [12]. To further illustrate the rise of citi-

zen science in practice, three professional organizations have been established: The Citizen Science Association (CSA) in the U.S., the European Citizen Science Association (ESCA) in Europe, and the Australian Citizen Science Association (ASCA) in Australia. A U.S. government website was established (<https://www.citizenscience.gov/>) to provide federal practitioners information about citizen science projects across the country. In addition, a new journal, *The Theory and Practice of Citizen Science*, was launched in 2016 [13]. Over the past 10 years, there has been a sharp rise in the number of peer-reviewed articles that have used or studied citizen science [14].

Citizen science projects can be classified by level of engagement of the participants in the science process [15]. At the less involved end of the spectrum, citizen scientists may contribute to data collection of a scientist-sponsored project. In "extreme citizen science", participants collaborate in the science process from problem definition, data collection and analysis [16]. The subject matter spans many different scientific disciplines such as astronomy, mapping the human brain, bird counts, rain data, and many, many more. However, citizen science engagement in public health is relatively limited, but has great potential to make significant contributions [17]. For example, researchers in Europe recently used citizen science approaches to detect large populations of the tiger mosquito, *Aedes albopictus* to prevent possible outbreaks of Chikungunya [18].

The benefits for citizen scientists are thought to be many and include: enhanced science literacy and knowledge; improved scientific thinking and ability to interpret scientific information; strengthened connections between people, nature and place; increased community-building, social capital, social learning and trust; and motivation by citizen scientists to influence policy and/or improve living environments [17] [19] [20].

The benefits for science researchers are well established, including the potential to gain spatial and temporal data that would not be collected otherwise. Nevertheless, ensuring data quality for research can be a challenge and steps need to be taken within the project design to assure sufficient data quality for project needs [21] [22].

3. System Overview

The system is composed of a suite of mobile and desktop applications. Each application plays a crucial role in the different phases of the system. Figure 1 illustrates the different phases. In Phase 1, we train an image classifier using images collected by our collaborators—entomologists and graduate students—who carefully labeled the images. Collaborators use a standalone application that we have developed to label their images. In Phase 2, citizen scientists use the system to upload images captured in the field and receive recognition results. Finally, in Phase 3, the system alerts experts

in infectious diseases to review images of insects that citizen scientists have uploaded. The system organizes the images presented for validation to experts, and posterior data analysis and visualization, using the prediction probabilities generated by the recognition module. Experts have access to all of the results produced by the recognition module, which allows them to confirm and oversee the whole recognition process.

At the end of Phase 3, our working prototype produces files that can be used by external analytics tools. Currently, we are loading these resulting files into a Google Maps script. However, future iterations of this software, intended for actual field work, will include a fourth phase, in which a more sophisticated statistical analysis of user-submitted images will be used to provide public health experts with real-time information about developing or ongoing outbreaks of dangerous mosquitoes.

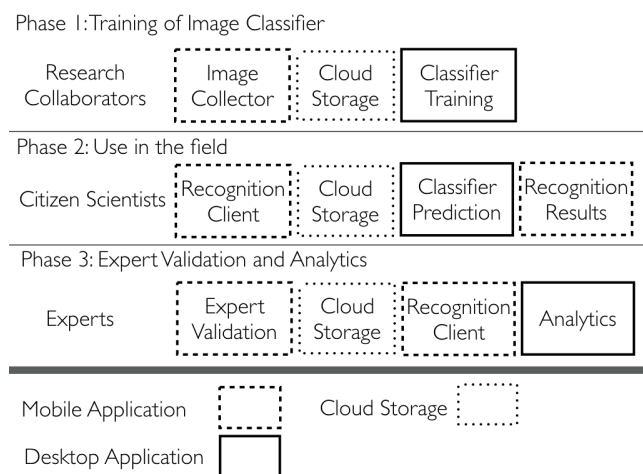


Figure 1: Phases of the system.

The different applications that compose the system streamline the processes of (1) collecting and labeling the images; (2) training classifiers with the collected data; and (3) using these models for recognition of images queried by citizen scientists. Raw image data as well as results of image analysis are stored in the cloud and are thus available to all modules of the system. This data takes the form of text files, or small binary files that store recognition results. Figure 2 illustrates the flow of information between the components of the system. We describe each of the applications and more details of the system in the following sections.

4. Working Prototype

In this section, we discuss the details of the prototype implemented as a proof-of-concept of the system. We begin our discussion with the process of training the classifier for the image recognition approach currently used by the system, and then discuss the set of applications that we have

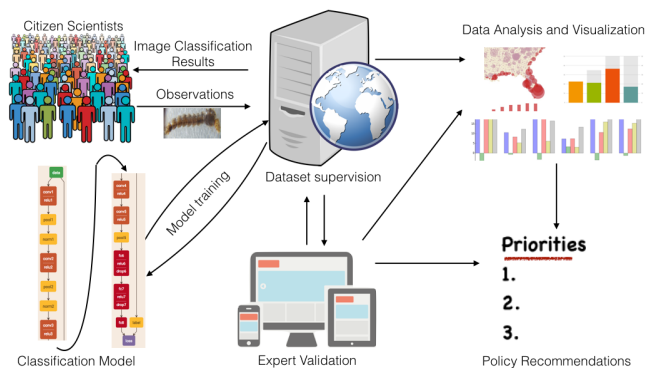


Figure 2: System Overview. A *Deep Learning* classification model is trained using images provided by our collaborators. Citizen Scientists interact with the system and can submit images for recognition. They receive results that indicate the probability of the presence of potential disease-carrying insects in their pictures. New, better classification models can be trained from new data received from citizen scientists or our collaborators. Experts have access to recognition results to oversee the system. Data collected can later be used in visualizations and in the proposal of public policies.

implemented for our first working prototype.

4.1. Training of Classifiers

The process of training the classifiers is a three-step process. First, we must collect training images—already labeled, e.g. as larva, pupa or mosquito—from collaborators or citizen scientists. Second, the system executes data augmentation and image processing routines on the collected images with the objective of increasing the variety and size of the training data. We describe the data augmentation and image processing steps later on in this document. Finally, the processed images are given as input to the *Deep Learning* network. Using these labeled training images, the neural network *learns* to recognize the images, adjusting its internal organization until it produces satisfactory results on a test dataset. At this point, the classifier has been trained, and we can make it available to the recognition module.

4.1.1. Image collection. *Deep Learning* approaches require an immense amount of training data. For instance, in the *ImageNet* challenge mentioned in Section 2, 1.2 million hand-labeled images are used for 1,000 different classes. We have developed a standalone mobile application that we call *Image Collector* used by collaborators to feed the system with training data. The application allows the user to label each submitted image. Images are either taken with the device’s camera or retrieved from the device’s storage. In the case of mosquito larvae or pupa, and as is illustrated in Figure 3, we

use a 60X Clip Light-emitting diode (LED) Microscope that can be attached to almost any smartphone.



Figure 3: 60X Clip Light-emitting diode (LED) Microscope attached to a collaborator's smartphone.

The application uploads the labeled image to our cloud storage, and that image is later used to train the recognition model. In an early phase of this project, some of our collaborators uploaded their pictures to a *Dropbox* [23] folder. However, this approach requires an extra step to manually label the received images. We see benefits in using the *Image Collector* application since it simplifies and streamlines the process of labeling images and feeding these images directly to train the classifier.

Figure 4 shows a screenshot of the data collection mobile application, *Image Collector*. Currently, this application has been tailored to those genera known to transmit diseases to people; consequently, the user can apply the following possible labels to the images:

- *Aedes*
- *Culex*
- *Anopheles*
- *Unknown*

Each image must also be labeled according to whether it is an *Egg*, *Larvae* or *Pupa*; and whether the image contains a *Head*, *Tail*, *Whole Body*, or if it is a *Side View* of the insect. Close-up photos of the head and tail are necessary to correctly identify the genus of the larvae.

If the collaborator submitting the image is not sure about the correct label, she can simply use the last option, “Unknown,” and leave the task of manually labeling the image to an expert. This application also collects the user name of our collaborator and a timestamp of the the image, allowing the system to keep track of each collaborator's contributions.

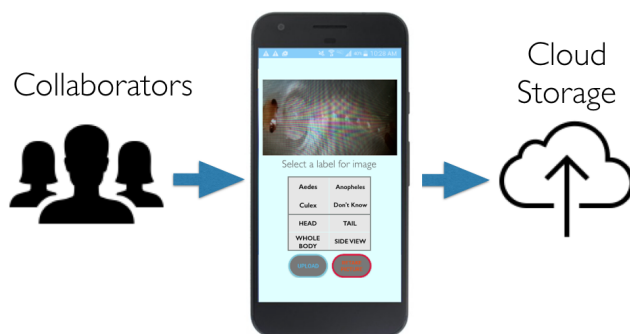


Figure 4: *Image Collector*: a standalone mobile application for image collection, labeling, and uploading to our cloud storage.

4.1.2. Image Processing and Data Augmentation To increase the variety in the data received by our collaborators, and to increase the size of the training dataset, we execute a data augmentation process on the collected images. First, each image received by our collaborators is manually processed to remove extraneous additional information contained in the image. For instance, borders are cropped since they do not usually contain any useful information for the classifier. The resulting cropped image is subject to the automatic data augmentation procedure that creates new images at different scales and orientations. Figure 5 shows an example of the data augmentation procedure. The transformed copies of the original image are generated using:

- Random rotation of the original image in a range of 0 to 60 degrees.
- Random horizontal and vertical shift in a range of 0 to 1/10 of the size of the image.
- Random shear in a range of 0.20 radians.
- Random flip either vertically or horizontally.

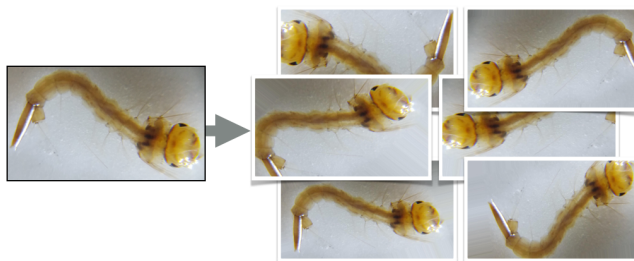


Figure 5: Data Augmentation Step. Each processed image is transformed to create multiple different versions of the original image. On the left is the original image, and on the right are samples of the new images generated from transforming the original image.

In addition, as shown in Figure 6, we perform histogram equalization to enhance the contrast of the images.

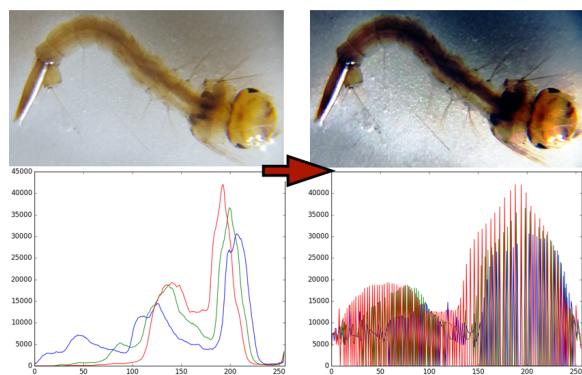


Figure 6: Histogram Equalization.

4.1.3. Model training Once we have increased the size of our training set of images, we proceed to train the *Deep Learning* network. We currently use the *CaffeNet* network from the *Caffe Deep Learning Framework* [24]. This model is an implementation of the *AlexNet* neural network, with some minor variations. *AlexNet* has historic importance in image classification, as it was the first *Deep Neural Network* to win the *ImageNet* classification competition in 2012. Details about the inner workings of our current selected neural network architecture can be found in [5]. *AlexNet* was developed to classify images based on the 1000 classes of the *ImageNet* competition. This means that this network is over-specified for our problem, since, for now, we only need to discriminate between 48 different classes. We are including, as future work, the simplification of the network to improve performance without losing accuracy.

We train the model using 80% of the augmented dataset (original plus transformed images). The remaining 20% of the images are used for validation of the model. Once we have a trained model, we proceed to deploy it in the system's recognition module, which is explained later on in this paper.

4.2. Citizen Scientist Mobile Application

The mobile application currently runs on any smartphone with the Android Operating System. The application first collects the citizen scientist's email to manage all communication between the system and the user. Once the system has collected this basic information from the participant, the citizen scientist can submit pictures, either by using a smartphone camera or by retrieving a picture previously stored on the device, similar to how collaborators share their pictures with the system during the step for training the image classifier. In both cases, the image is sent to the system's cloud storage.

In addition to the image, a text file is transmitted to the cloud storage. This text file contains additional information, including the geolocation of the image. Using this text file, as opposed to encoding this meta-information in the image, helps maintain a separation between images and the associ-

ated user. Not only is this process helpful for implementing user privacy settings, but it will also allow the system to receive other kinds of information from the user—e.g., additional descriptions of the environment where the image was collected, either collected from the device's sensors or manually entered by the citizen scientist—without altering the image's metadata.

After the user has submitted its picture to our cloud server, the system will send a confirmation that the image was received. The user can now proceed to open another view in the application where the recognition results will be displayed. These results are retrieved from the cloud by the user's application. Figure 7 shows a screenshot of this functionality. The recognition results include the top classes detected by the system and the corresponding confidence probability of the classification.

Currently, the user's email is the only identifier used by the system, but in future versions of the system, a random unique ID might be generated for each new user.

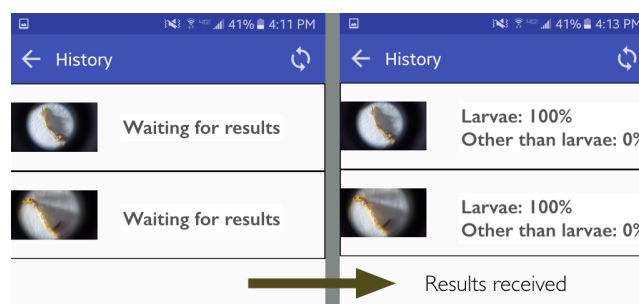


Figure 7: Screenshot of the list of images submitted by the user with their respective recognition result.

4.2.1. Overlay guidelines to assist the end user in taking a picture. To further assist the user when taking the picture of the specimen, we have included an overlay with transparencies that guide the user and assist her positioning the smartphone. In the particular case of mosquito-borne diseases, and as shown in Figure 8, the overlays include pictures of eggs, larvae and pupa obtained from the *Centers for Disease Control and Prevention (CDC)* website. The purpose of the overlay feature is to improve the quality of the image taken by the user, and also to improve the positioning of the specimen to be recognized. The overlays implemented in our prototype range from a simple *water line* to help the user align the specimen if hanging on water and on side view, to more complex overlays, such as mosquito species in different stages (*egg, larvae, or pupa*). (The overlays are *not* part of the image, and are not transmitted to cloud storage.)

4.3. Recognition Module

The recognition module has been conceived to be portable and efficient. We envision this module to run on

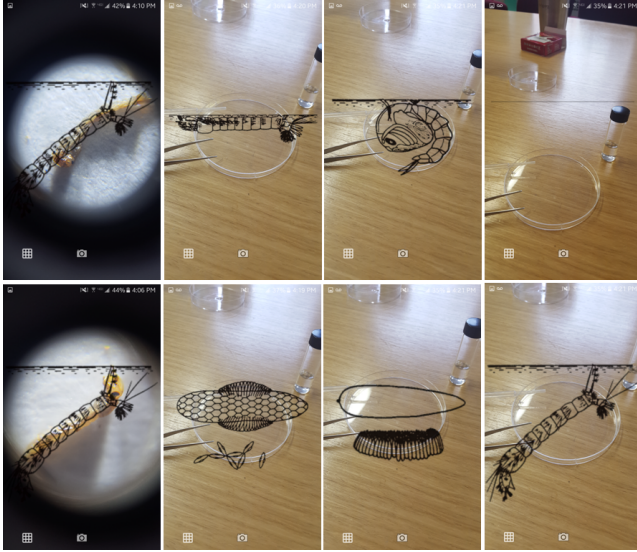


Figure 8: Example of the overlays used to assist the user when taking a picture.

a workstation by the manager of the system. In the case of an outbreak, we want the manager of the system to be able to utilize this module anywhere she has an Internet connection. This component interacts with our cloud storage. It retrieves new images that have been uploaded for recognition by citizen scientists and uses the trained *Convolutional Neural Network (CNN)*, the classifier from Section 4.1.3, to predict its possible class.

Once the recognition module has processed a new image, it uploads its predictions to the cloud so the end user's application can access the prediction results. The recognition module also maintains a copy of the results locally so that they can be used later for analytics without the need for the component to be online.

The approach that we have taken to implement this recognition module makes it very easy to upgrade the classifier. Once the system has received new images, either from our research collaborators or citizen scientists, we can retrain the classifier to improve its prediction accuracy. Existing results of images already submitted by end users can be reprocessed to provide a more accurate label. Since the classifier runs on the manager's computer, we can quickly deploy the upgraded classifier to obtain better predictions immediately.

4.4. Expert Validation

Despite all the recent successes in image classification, there is always the possibility of an erroneous classification [25]. To confront this issue, we anticipate the use of human experts to monitor the recognition results of the system. We have created a mobile application that we term *Expert Validator*. This application connects to the cloud storage and allows experts to review the classification results.

Images are presented to the expert, sorted by the confidence value of the classification. The user can click on the image to get a complete set of results if the list of results is too large. Figure 9 shows a screenshot of the image verification process. First, the images submitted by end users are marked as *pending validation*. Second, experts confirm that the label assigned to the images is correct. Finally, the end user is notified that an expert has validated the image.

On the end user's application, images that have not undergone validation show a *Pending Validation* label next to the prediction results. Those images that have been validated by an expert show either the *Accepted* or *Rejected* label next to the prediction results.



Figure 9: Top: End user interface showing two images that have not been verified by an expert. Middle: The *Expert Validator* application is used by experts to verify images that have been flagged by the system. Bottom: The user is notified whether a submitted image has been accepted or rejected by an expert, despite of its recognition result.

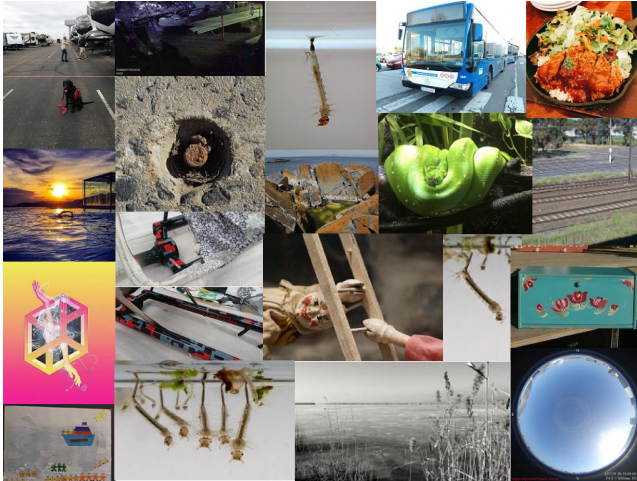


Figure 10: Samples from the test dataset downloaded from the Internet.

5. Pilot Evaluation of the First Prototype

We tested the system to determine its recognition performance on random images obtained online. We also tested the whole application pipeline simulating an outbreak in the vicinity of our campus. We used a Dell Precision 5510 laptop computer with 8GiB of RAM, an Intel Core i7-6700HQ processor and an Nvidia Quadro M1000M GPU. The recognition module runs on Ubuntu 14.04. The mobile components of the system were tested using three Android smartphones: a Samsung Galaxy S5, a Samsung Galaxy S6, and a Google Pixel.

We first tested the recognition performance of our classifier to discriminate between larvae and non-larvae. We used a random set of 550 images from the website *Flickr* [26]. These images were obtained by implementing a web crawler that downloaded random images of anything other than larvae, and 100 random images with the words “mosquito larvae” in their description. Figure 10 shows some examples of the images in this dataset. The images were then labeled to indicate whether they were considered to be *Larvae* or *Other than larvae* based on the label assigned by each *Flickr* user. We trained a classifier using the images received by our research collaborators in other labs and universities. We then ran the trained recognition module on the Flickr images. The classifier produced the following results: Precision 100% and Recall 47.4%.

One hundred percent precision means that every single positive prediction that an image contained mosquito larvae was correct, i.e., the system's prediction did not include any false positives. The system did, however, have a few false negatives, where the system did not detect some images containing larvae. Of the set of 550 images collected randomly from the web, 57 images contained mosquito larvae. The recognition module recognized 27 of these 57 images or 47.4

Execution time	
Procedure	Milliseconds
ONE TIME	
Setup Classifier	3940
PER IMAGE	
Load Image	0.94
Resize Image	1.30
Histogram Equalization	3.24
Prediction	6.47
– (Forward Pass - GPU enabled)	

Table 1: Execution time of procedures in the recognition module.

percent.

We anticipate that in future iterations of the system, with further training, the recall value will be greatly improved. The current recall rate is primarily due to the fact that the random images in the online dataset contained features that had not been included in the initial training process. For example, the system was not trained to recognize images with multiple, sometimes overlapping, specimens in the picture. Figure 11 show examples of these images. The recognition process was unable to recognize the arrangement and quantity of the specimens in the picture as containing larvae. We are currently exploring the use of object proposal algorithms, e.g., [27] [28], to help the system improve its performance in these kinds of images. It is important here to consider that citizen scientists will use the overlay guidelines to take better pictures than the images gathered randomly from the Internet to test our first prototype. In any case, the precision and recall values obtained from this dataset are interesting when looked at from the perspective that they reflect the behavior of the recognition module in the “wild”.

Table 1 lists the execution time of procedures executed in the recognition component. Once the classifier is running, it can process approximately 83 images per second in GPU enabled mode.

5.1. System Response to Low Quality Images

Based on our tests, we expect that if a citizen scientist uploads low quality images, the recognition module, which has been trained using high-quality labeled images, will reject them by giving them a low probability value for the class in question. The second line of defense is in the expert validation phase, in which experts could reject low quality images that have been flagged by the system.

5.2. Outbreak Simulation and Sample Visualization of Geographical Regions of Interest

In the event of an outbreak, we expect citizen scientists to upload their observations and receive recognition results



Figure 11: Examples of false negatives. Larvae is present in the images but the classifier does not recognize them as containing larvae, likely due to the arrangement and quantity of the specimens.

in return. The recognition module will be continuously updated with every batch of images received from citizen scientists. Epidemiologists will get precise information about geographic regions where relevant specimens have been spotted. This information will help them organize a response to the crisis. The system will help experts filter information received from citizen scientists, and this filtering will help them efficiently process more information than if they had to manually classify the images received from the users.

We tested the system in the vicinity of Brooklyn College. Volunteers walked around campus with the end user application installed on their smartphones. These volunteers chose places where mosquito larvae could be present in warmer months. Instead of using pictures taken from the camera on the device, they used images that were previously stored on the devices. The end-user applications uploaded the images to the cloud server and these images were later classified by the recognition module. The system correctly identified the images that contained larvae and passed the information to a Google Maps visualization script. Figure 13 shows a visualization produced from the exploration of the volunteers. Further analytics can be performed with the data collected (geolocation and classification results). The modularity of the system makes it easier to streamline the resulting data to other visualization tools, e.g., Google Maps. In the event of a real outbreak, experts could use these types of visualizations to identify regions that may require urgent attention from public health experts.



Figure 12: True Positives. Images correctly identified as mosquito larvae by the system.

6. Discussion and Future Work

As exciting as we find this prototype, we believe that the system can further be improved with contributions from external developers. We have made our software public so others can continue building on the the work we have done. Below are aspects of the system where we would like to see further improvements.

a. Deploying the model to the end user's smartphone.

Deploying the model and executing the classification algorithms on the user's side remains under consideration. This approach would result in an immediate recognition of the image, but it would also add more complexity when deploying improved versions of the classification model. For instance, when the classifier is upgraded, we would have to push the model to each device using our application. Currently the model takes 228 Megabytes which could be a storage burden for old smartphones. On the other hand, in the current configuration of the system, if we have a centralized location for the model, i.e., recognition module, we can continuously upgrade it without affecting the end users or their limited devices.

b. Addition of complementary Computer Vision algorithms.

The nature of the problem of recognizing species of insects in images requires other approaches that are specific to the behavior of these insects and how that behavior is captured by the images. We believe that using other complementary approaches to image segmentation and recognition can refine the suggested classifications obtained from the *Deep Learning* algorithms. Another idea is to include the collection of video in the citizen scientists' applications, which can allow us to determine the species of mosquito based on their movements.

We have considered including an *object proposal* algorithm,

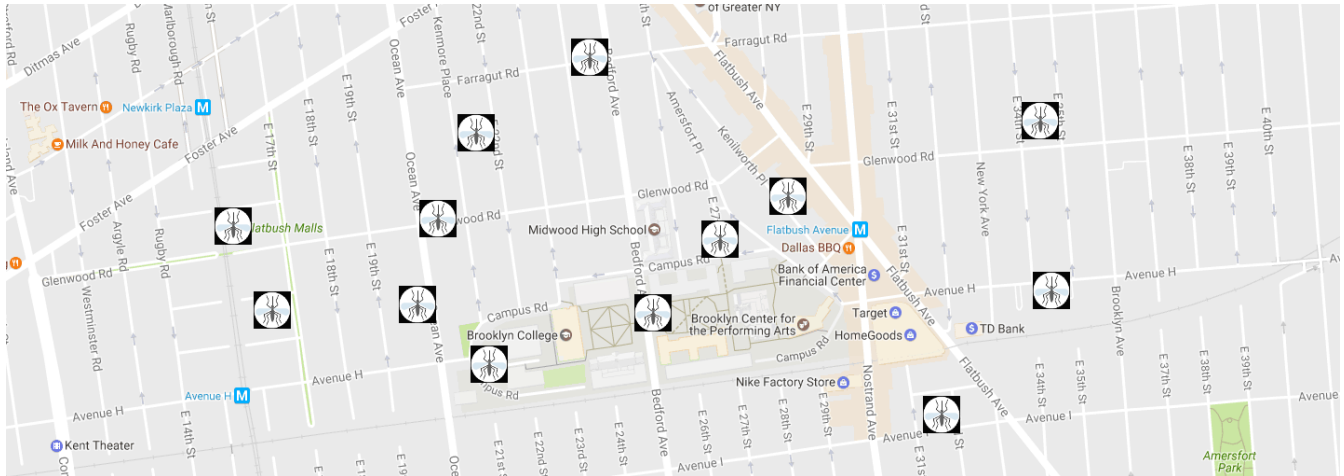


Figure 13: Testing the system and visualizing the locations of the positive recognition results.

e.g., [27] [28], in future iterations of the system to facilitate the selection of the specimens on the training images. These algorithms analyze the image and suggest possible locations of objects. These approaches use features in the images, e.g., edges, saliency, or color variation, to make predictions about the location of objects in images.

c. Simplification of current Deep Learning network. The current architecture used by our recognition module, *Caffenet/AlexNet*, was developed to classify images based on the 1000 classes of the ImageNet competition. This means that this network is over-specified for our problem since, for now, we would only need to discriminate between a few different classes. Future work will include the simplification of the network to meet the needs of our problem. This reduction in the complexity of the network will improve performance without sacrificing accuracy.

d. Multiplatform end-user application. Future work includes expanding the supported software platforms for the system. Even though the Android Operating system is used by a large number of end users, we believe that our prototype must also include versions that run on other operating systems, including iOS, the operating system used by Apple devices.

e. Privacy Considerations. It is important to mention that the goal at this stage of our research is to use the prototype as a proof-of-concept. We must develop a more secure and robust system in future development iterations, particularly if the proposed system is going to be deployed on a large scale. A system deployed to the public must make sure that any information that can possibly link images to the user must be very carefully handled, particularly, when geolocation is included. Future work must include the generation of secure user IDs. Actions must be taken to prevent malicious people from gathering data from citizen scientists.

f. Providing additional information to the user. In addition to recognition results, citizen scientists could receive

other information that is tailored to the recognition results and the disease that the system is helping to combat in the future on their mobile applications.

g. Training the system with images that the system may recognize as false positives. To make the recognition module more accurate, we are currently collecting images of Chironomidae Larvae, Oligochaeta and Planaria. Since these specimens can sometimes be confused for mosquito larvae, we plan to train the system using those images as negative samples. We hope to make the system more accurate and reduce misclassifications.

h. Web Interface for Citizen Scientists. We are currently developing a web-based application on top of the open source *Caffe* web interface [24]. This web interface will allow the end user to upload an image to our server and get an immediate classification result. We envision that if the system is released to the public, this application could have a significant impact in the case of an epidemic outbreak, particularly if the website contains useful information about the epidemic in question in addition to accessing the recognition component.

i. Real Time Analytics and Management Applications. In the case of an outbreak, the system could benefit from new applications that display real time analytics of the state of the system. More sophisticated Graphical User Interfaces (GUI) can also be developed to better assist the users of the system.

j. Large-scale simulation of an outbreak. Future iterations of the system must be evaluated to determine its capacity to handle a large number of submissions from citizen scientists. This evaluation will allow us to better estimate the necessary amount of computing power required and the level of organizational and governmental commitment in case of an outbreak. This evaluation must also include varying configurations for the number of image uploads per user with the goal of making the system more robust.

7 Conclusion

Current state-of-the-art technology allows for high accuracy classification of images. In this paper, we have presented a modular system and its first prototype that is capable of leveraging Computer Vision techniques to empower communities affected by an arbovirus epidemic. The proposed system can provide valuable information to experts in charge of coordinating solutions. Our modular system provides a suite of applications that are capable of recognizing mosquito larvae in images with great accuracy. Future iterations of the system can be deployed to the citizen science community and help in the fight to control potential epidemics.

A video demonstration of the prototype and links to source code are available at www.assistiverobot.org/system. We expect that by opening these applications to the research community, the system will undergo a continuous process of improvement.

Acknowledgments

This work was supported by NSF EAGER #1645154 and NASA #NNX15AG02G. We would like to thank all our collaborators, particularly Krishna Woerheide, for providing us with hundreds of pictures of mosquito larvae, and the NASA's GLOBE Observer team for their feedback.

References

- [1] A. S. Fauci, D. M. Morens, and N. E. W. Engla, "Zika Virus in the Americas Yet Another Arbovirus Threat," *New England Journal of Medicine*, vol. 374, no. 7, pp. 601–604, 2016.
- [2] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, mar 2004.
- [3] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, "Action Design Research," *MIS Q.*, vol. 35, no. 1, pp. 37–56, mar 2011.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information and Processing Systems (NIPS)*, pp. 1–9, 2012.
- [6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [8] R. R. Schaller, "Moore's law: past, present and future," *IEEE Spectrum*, vol. 34, no. 6, pp. 52–59, jun 1997.
- [9] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout : A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014.
- [10] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," *ICML*, vol. 32, pp. 647–655, 2014.
- [11] R. Bonney, T. B. Phillips, H. L. Ballard, and J. W. Enck, "Can citizen science enhance public understanding of science?" *Public Understanding of Science*, vol. 25, no. 1, pp. 2–16, 2016.
- [12] "citizen.science @ en.oxforddictionaries.com."
- [13] R. Bonney, C. B. Cooper, and H. Ballard, "The theory and practice of citizen science: Launching a new journal," *Citizen science: Theory and practice*, vol. 1, no. 1, pp. 1–1, 2016.
- [14] D. C. McKinley, A. J. Miller-Rushing, H. L. Ballard, R. Bonney, H. Brown, D. M. Evans, R. A. French, J. K. Parrish, T. B. Phillips, S. F. Ryan, and Others, "Investing in citizen science can improve natural resource management and environmental protection," *Issues in Ecology*, vol. 19, 2015.
- [15] R. Bonney, H. Ballard, R. Jordan, E. McCallie, T. Phillips, J. Shirk, and C. C. Wilderman, "Public Participation in Scientific Research: Defining the Field and Assessing Its Potential for Informal Science Education. A CAISE Inquiry Group Report," *A CAISE Inquiry Group Report*, no. July, p. 58, 2009.
- [16] M. Haklay, *Citizen Science and Volunteered Geographic Information: Overview and Typology of Participation*. Dordrecht: Springer Netherlands, 2013, pp. 105–122.
- [17] L. Den Broeder, J. Devilee, H. Van Oers, A. J. Schuit, and A. Wage-makers, "Citizen Science for public health," *Health Promotion International*, p. daw086, 2016.
- [18] J. R. B. Palmer, A. Oltra, S. Mariani, S. Escartin, E. Pous, S. Vergony, E. Marqu, and C. Vall, "Preparing for Chikungunya : Using ovitraps , citizen science , and human mobility data to improve epidemiological models in Northeastern Spain," pp. 1–23, 2015.
- [19] B. K. Haywood, "A Sense of Place in Public Participation in Scientific Research," *Science Education*, vol. 98, no. 1, pp. 64–83, 2014.
- [20] A. C. King, S. J. Winter, J. L. Sheats, L. G. Rosas, M. P. Buman, D. Salvo, N. M. Rodriguez, R. A. Seguin, M. Moran, R. Garber, B. Broderick, S. G. Zieff, O. L. Sarmiento, S. A. Gonzalez, A. Banchoff, and J. R. Dommarco, "Leveraging Citizen Science and Information Technology for Population Physical Activity Promotion," *Translational Journal of the American College of Sports Medicine*, vol. 1, no. 4, 2016.
- [21] R. Bonney, J. L. Shirk, T. B. Phillips, A. Wiggins, H. L. Ballard, A. J. Miller-Rushing, and J. K. Parrish, "Next Steps for Citizen Science," *Science*, vol. 343, no. 6178, pp. 1436–1437, 2014.
- [22] J. Hunter, A. Alabri, and C. van Ingen, "Assessing the quality and trustworthiness of citizen science data," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 4, pp. 454–466, 2013.
- [23] "Dropbox." [Online]. Available: www.dropbox.com
- [24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675–678.
- [25] A. Nguyen, J. Yosinski, J. Clune, A. Nguyen, J. Yosinski, and J. Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015.
- [26] "www.flickr.com."
- [27] P. Dollar and C. L. Zitnick, "Structured Forests for Fast Edge Detection," *2013 IEEE International Conference on Computer Vision*, pp. 1841–1848, 2013.
- [28] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 391–405, 2014.